SRPeek: A Super Resolution Based Screen Peeking Threat Model

Jialuo Du
Tsinghua University
dujluo@gmail.com

Zhichao Cao
Tsinghua University
caozc@msu.edu

ABSTRACT

We use smartphones everywhere and concerns have arisen about ‘shoulder surfing’, or the act of strangers peeking at our phone in public areas. This privacy threat have been studied heavily and various countermeasures have been designed, but mainly against attackers with the naked eye. With the development of smartphones and super resolution (SR) technology, however, we believe that this threat have been underestimated, as a malicious observer can attack from farther away with the assist of his smartphone camera and SR algorithms. We underline this concern by presenting SRPeek, an end-to-end system of shoulder surfing, containing a unique multi-image SR neural network architecture designed for this task, which in this application outperforms other known SR algorithms and fills the blank space of multi-image SR. We prove that our attack is efficient, allowing a human to read 95% of the text off a smartphone screen at 1.5m range (in the experiment’s circumstances) and is robust in different environments. Our SR network can also be used in word recognition tasks with less than ideal images as a preprocessor.

1 INTRODUCTION

As already we are aware, smartphones have become a necessity in our lives, as we are checking our mobile phones constantly throughout the day. As a result, some concerns of privacy have arisen about nearby parties peeking at our screen, or ‘shoulder surfing’. Although great effort has been put forward to mitigate this threat, from physical privacy films to alternative password entry interfaces [27] [20] and input methods [13], these methods often require additional cost and/or effort [4] and are not widely implemented yet.

On the other hand, many works in this area are built on a threat model of an attacker observing with his/hers naked eye, as they assume that the attacker is not malicious and merely take several peeks out of curiosity. However, it is the rare but malicious and prepared attacker that can do the most harm, and leave aside special equipment, if we only equip the observer with a smartphone, he/she can not only acquire sensitive information from long distances, reducing suspicion, but also record the information for propagating, dealing greater damage to the victim. For example, in a Senate hearing, the Justice Secretary of Philippines, Vitaliano Aguirre II, suffered a leakage of his text messages, as someone had taken a snapshot of his smartphone [21]. Moreover, smartphone cameras have seen great improvement these years, and as a highly-accessible ‘extension of the eye’, it is possible that ‘shoulder surfers’ peek at others’ screen through their smartphone lenses to obtain critical information like passwords or private e-mails. What’s more, recent developments in the field of super resolution (SR) pose a greater threat to smartphone privacy. Attackers can take multiple snapshots of the victim’s screen and process them with multiple image super resolution algorithms in real time, making them able to see clearly while keeping a longer distance. However, most SR methods are designed for and trained on naturally taken images [19] [17]; few works have focused on reconstructing blurry snapshots taken at extreme distances, and a new architecture needs to be designed.

To explore this rarely studied privacy threat of the attacker looking through a smartphone camera, we have developed a holistic system for shoulder surfing. The fictional attacker captures multiple images of the victim’s screen in burst mode from his smartphone, and enhances its resolution with our multi-frame super resolution neural network. The network is designed and trained for this purpose, targeting several blurry images taken in quick succession of a screen at a distance. As the information of interest is mostly displayed on screen in the form of text (e.g. passwords or e-mails), our network focuses on reconstructing text, specifically, Chinese characters, English letters, and numbers. The system is deployed on a smartphone for demonstration, capturing and processing images at real time. Our system can also be used as a preprocessor for text-recognition applications when multiple images are available, such as real-time translation apps on a smartphone or text-recognition of video clips.

1.1 Difficulties

There are several difficulties unique to this SR application. The most prominent one is the blurriness, as our photos are taken at extreme range, taken secretly and hurriedly without much room for focusing, while straining the magnification of the smartphone lenses. Unlike most SR applications and datasets working with ‘normally’ captured photos, the images we face exhibit lower concentrations of information and more artifacts, and needs more ‘reconstruction’ than ‘interpolation’.

Another difficulty unique to our task is the subject—mainly, Chinese characters. Composed of multiple strokes, these characters are distributed discretely in image space, not known to other entities, e.g. faces and natural objects. In some cases, messing up a stroke will not impact its readability, and in other cases shortening or lengthening a stroke even slightly will lead to a different character, leading to misunderstandings. And unlike most SR applications working on similarity (between their output and the ‘ground truth’), our goal is readability, and the network architecture and training process must be engineered accordingly.

1.2 Network Architecture

As mentioned above, reconstructing images with high degrees of blurriness requires a unique approach. Most works on multi-frame SR function on video clips [16] or multiple snapshots [28], however, our application is subtly different from the two. Our dilemma is as follows: on one hand, each one of the photos we are to process is blurred with a randomly different PSF kernel, which, because of the extreme blurriness, cannot be approximated as a constant, isotropic

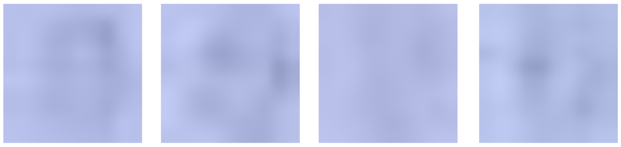


Figure 1: Fig 1. 4 photos of number '0' on a screen at a distance of 1.5m, taken in quick succession from a still smartphone camera. Note how they exhibit different blurriness and display no consistency between neighboring frames.

gaussian kernel, and thus lacking consistency between neighboring frames, meaning that they cannot be processed as a video clip(see Figure 1); on the other hand, because of the low concentration of information, the blurred photos are similar to pieces of a jigsaw puzzle, each containing only fractions of information, and the only chance of recovering the ground truth is by comparing each photo against others. If processed by common procedures enhancing the resolution of a series of snapshots—processing each one separately and merging them afterwards, few useful information can be extracted from each one of the images, and merging them will not produce satisfactory results.

Considering the unique challenges of our application, we believe that if these images are processed iteratively, alternating between the following two processes, our network will be able to solve this 'jigsaw puzzle':

(1) Every single image of the input collection will be processed solely (by several layers of CNN), with access of the output of step (2) of the last iteration;
(2) The processed results will be merged (with weighted averaging methods) and distributed to each photo for the next iteration.

This architecture is beneficial to our task. On one hand, the deep learning part is assigned to each single image, reducing computational complexity and parameter growth, while receiving a global view of all the images, renewed at each layer, thanks to the merging process; on the other hand, information can be shared horizontally with weighted averaging methods, meaning no consideration of sequential order and no reliance of consistency between neighboring frames. Thus, this architecture solves the dilemma mentioned above, and proves perfectly functional in our application.

The details of our architecture will be discussed in section 4.

This paper makes the following contributions:

- We propose SRPeek, a multi-frame SR neural network architecture aiming at reconstructing extremely blurred and defocused images. This model is not only functional in our scenario, as a shoulder-surfing threat model, but also can be used in text-recognition applications prior to recognition algorithms to increase accuracy.
- We design a threat model of shoulder-surfing, with the attacker armed with multi-frame SR algorithms and taking multiple photos (in burst mode) of the victim's screen with a smartphone camera. To the best of our knowledge, we are the first to consider the presence of smartphone cameras and SR algorithms in shouler surfing senarios.
- We demonstrate the effect of this shoulder-surfing attack and prove that it poses a threat to screen privacy.

The rest of the paper is organized as follows. Section 2 describes the threat model of our carefully studies the performance of state-of-the-art concurrent flooding. The detailed design of COFlood protocol is shown in Section 3. We show the implementation details and evaluation results in Section 4. The related work is introduced in Section 8. Finally, we conclude our work in Section 9.

2 THREAT MODEL

The objective of the attacker is to acquire the information on the smartphone screen of the victim, reading the numbers and English/Chinese characters on screen, with the help of his smartphone's camera equipped with the SR network we designed. To reduce suspicion, the attacker will probably position himself at about 1.5 meters from the target screen, at an angle within 30 degrees. At this distance few people will be on guard of strangers even when viewing sensitive data or entering passwords. The attacker may raise his phone (one of the latest models with powerful lenses and computational abilities), pretending to be interacting with it, while pointing his camera at the victim. He will extend his zooming ability to the maximum, try to get a focus on the screen (which is difficult to achieve), and take 20 to 30 images in burst mode, with about 0.05 0.1 seconds of interval between frames. This process will take about 2 seconds, and we assume the information on the screen will not change in this period, nor the position of the screen. These images will be fed to a multi-frame SR network and the result, one single image with higher resolution, will be displayed on the screen soon afterwards. The characters in the image will be reconstructed to the best of the network's ability, and if successful, the attacker will be able to decipher the information. The above procedure can be automated by an app and repeated every few seconds, giving the attacker a continuous surveillance over the victim's screen.

3 SYSTEM DESIGN

We designed an end-to-end network architecture consisting of several identical layers in which: Each input image (or each lane from the output of the previous layer) is assigned to its own lane and processed by several convolutional layers. These layers share parameters across channels; however, each image is processed individually and do not interact with each other in this stage. Each channel will present a feature map of n channels. The features extracted from all layers are merged broadwise with the following function:

$$x'_k = \sum x_i e^{kx_i} / \sum e^{kx_i}$$

Where k varies from -1 to 1. Apparently, k=0 leads to averaging, $k=+\infty$ leads to a process similar to max operator and $k=-\infty$ leads to min operator. This process merges information from every lane into one. The output is n channels for each value of k. The original input images are accessed again. These images are also processed separately by several convolutional layers (sharing parameters across lanes) to extract features. The output of these images (m channels each) are stacked with the n channels from step 2) as m+n channels each lane. Some more convolution and an upscaling layer may follow, merging the data more thoroughly. These will be the input for the next layer. The above 3 steps are

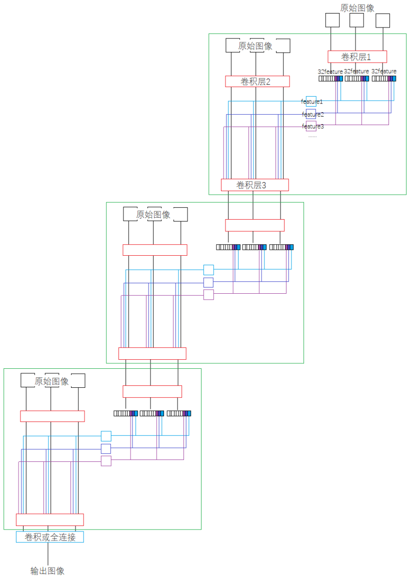


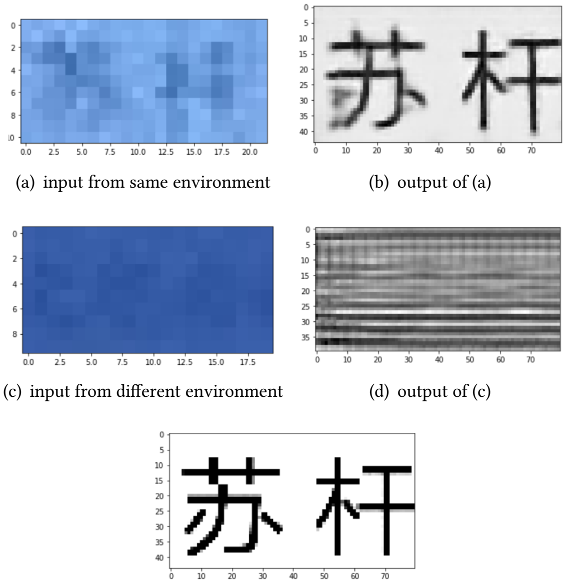
Figure 2: Network Architecture of SRPeek.

stacked as one layer. The functional network often consists of 5 to 8 layers to achieve 4x super resolution. The network ends with a merging process similar to the 2) step and a few more layers of convolution. The full structure of a 3 layer network is shown in Fig 2.

4 IMPLEMENTATION AND TRAINING

4.1 Experiment Setting

In our experiment, we are forced to collect the training dataset on our own instead of using public datasets. Because of our unique application, to the best of our knowledge there is no publicly available image dataset built for shoulder-surfing. Modern network architectures work with naturally obtained images, e.g. ordinary photos, satellite images, scanned documents, YouTube videos, etc. where the images are well focused and captured within the imaging ability of the camera, with the objects of interest, e.g. the face or the printed word, at least visible to the naked eye. and we apply the SR algorithms to reduce noise and refine details, like repainting texture and refining edges. However, as emphasized above, in our application we face images that is extremely blurred and distorted, due to the extreme circumstances when the photos are taken, and it is impossible to tell from each single picture whether a stroke of a character really belongs here, as it's equally possible that this line is a distorted mixture of multiple lines or is a mutilated part of a longer stroke. Noise and details are not the main issue here, as we need only the facts, deeming it a victory if the neural network returns a noised, wrapped, but readable image. These are the



(a) input from same environment

(b) output of (a)

(c) input from different environment

(d) output of (c)

(e) ground truth

Figure 3: Results of training with data collected in fixed environment. The network performs well in this environment but fails in other.

reasons why our work does not choose datasets that are publicly available, and collect data by ourselves–as it will completely defeat the purpose if we turn to these datasets. And this also explains why we failed in trying to get comparable results with other SR architectures we use as baselines, as these networks are designed for another purpose and they hardly ever faced such distorted and blurred images.

The task of data collection is quite tedious, as we discovered that we need photos taken in all kinds of environments to train a robust network. We initially collect 60,000 images with the position of the smartphones static and all other environmental parameters stationary, and divided it into training and testing datasets which do not have common characters between them, so that in the testing phase the network has to recover characters that never appeared in the training dataset. Our network easily learned the patterns and presented equally satisfactory results in the test dataset (see Fig 3(b)), displaying clean, easy to read results, showing that the network has learned to extract features beneath the character level, and it is highly possible that this network can recover all kinds of characters, not limited to the characters in the training dataset. However, when we slightly modified the environment, the images displaying the text with a different shade and size, none of the features were successfully extracted, and the network outputted white noise (see Fig 3(d)).

As a result of these observations, we have to collect images in varied environments. Our experiment consists of 2 smartphones at a distance between 1 and 2 meters, one for the attacker and one the victim. An app runs inside the attacker's phone, taking photos continuously, adjusting its focus and aperture, trying it's best to capture high-quality photos. Another app runs inside the victim's phone, displaying random characters with several fonts and colors, for the attacker to capture. The characters are selected from commonly used Chinese characters with 5 to 10 strokes and the English alphabet, and we divide this assemble into training and testing subsets. As the English characters are apparently easier to classify and reconstruct for the networks, the experiments testing the performance of the network will be performed only on Chinese characters, but it is our observation that our system works on English characters as well as Chinese characters. As the experiment goes on, the attacker phone will keep on taking photos in burst mode (20 photos per time), and the victim phone will change the characters it is displaying at a fixed interval, when approximately 100 images were taken by the attacker. We change the environmental setting whenever about 2k photos were taken, relocating the phones (while keeping their distance between 1 and 2 meters) or modifying the angle of the phones. The attacker will always point its phone directly at the victim, keeping its screen in the middle of the photos; The victim, however, may tilt its phone at an angle within 30 degrees. We then readjust the focus and aperture of the attacker's phone to maximize the image quality before continuing data collection. Screenshots were taken in the victim's phone each time it changes its display, and these screenshots are scaled, spun and deformed according to the distance and angle between the phones at the time, and used as ground truth for training and evaluation.

The data was collected indoors, as is often the case of shoulder-surfing in public areas (theaters, subways, offices, etc.). We perform the experiment in a room with a window and repeat the data collection phase in different times of the day and night, with curtains drawn or open, lights turned on and off to modify the illumination parameter. The position of the phones is also a crucial factor in this parameter, and we avoid having the sunlight directly shone on the camera of the attacker or the screen of the victim, as it is extremely difficult to see anything in these extreme circumstances. We also draw a box around the characters on the victim's screen for alignment purposes, as although in production our system can detect the edges of the screen for alignment, this method provides more accurate results and possibly accelerates training. After collecting the images, we crop them along these boxes, retaining only the characters. In the training phase we randomly crop out squares of 9x9 pixels at the corresponding position on the training dataset and ground truth, and feed them into the neural network. We collected 800,000 images in this way with a Redmi6A phone (which possess a camera with 130 million pixels). This process is extremely time-consuming, but the data amount is crucial in our experiment, as slight variations of the environment will cause drastic changes in the features extracted from the characters, and only by covering the variations in each of the environmental parameters in the training data can the system successfully function in different scenarios.

4.2 Model Specifications

It is our observation that cameras in different phones display different patterns of distortion when performing the shoulder-surfing experiment, and apparently, images captured from a phone with weaker abilities (less pixels, less range of focus, worn lenses, etc.) will need a stronger, more complex model to extract the features. Thus, the specifications of our model are only for reference and may not work when reproduced on another phone.

Our model accepts 20 images at a time, with a size of 9x9 pixels. The model consists of 5 blocks described in the previous section. In each block, feature maps from the previous layer are passed through a single convolutional layer with 32 channels of feature maps as output. the 20x32 feature maps are then merged horizontally with the max-min-average process, and the output is 5x32 channels. Simultaneously, the original input images are processed again with a single convolutional layer with 32 channels output, and stacked with the former 5x32 channels to form a dataflow of 6x32 channels for each one of the 20 images. These channels are finally passed through a single convolutional layer, outputting 32 channels per image for the next block. The kernels in each convolutional layer is 3x3. We also insert LeakyReLU and Batch Normalization processes after each convolutional layer, and 2 upsampling layers between the 5 blocks. A single 1x1 convolutional layer is placed after the 5 blocks with a single channel as output to form the final output layer. This model consists of approximately 200,000 parameters and a complexity of about 400,000 FLOPs. As a light-weighted model, it makes a prediction within 0.1 seconds on a Tesla K80 GPU over a 9x9 patch, and when implemented on a smartphone, the human user can recognize a character within 2 seconds of processing time.

4.3 Training Process

Because of the difficulties in discovering the patterns among the blurry images, the training process is not so straightforward and somewhat time-consuming. Our approach is as follows: we first collect 60,000 images at a stationary experiment setting and train the model with it, as mentioned in Sec 4.1, until we get satisfactory results on the test dataset. The model should be able to handle this task easily. After that, we use transfer learning methods to fine tune the model in order to fit different environmental parameters. We add a small percentage of image data from another similar experiment setting into the training data and resume training. When the model stabilizes, continue adding data from the same setting until the model can equally process data from the two image collections. Note that in this process the model will tend to extract false features, displaying wrong but clear texts as result. This phenomenon can be mitigated with dropout and normalization layers. As the model successfully fits two of the image sets, we repeat this procedure for several times until it shows signs of self-adapting, for example, when presented photos taken at 2.0m and 1.5m range in the training process, the model fits 1.75m photos easily. After that, the model can learn with the full fully-shuffled dataset with fewer difficulties.

5 MODEL EVALUATION

We evaluate the ability of the network model, testing its performance in different training and testing conditions.

5.1 Performance In Controlled Environment

We train and test the model with the images captured with exactly the same environment parameters, and used Peak Signal to Noise Ratio to evaluate the accuracy of the recovered images(see

Environment	Day,1.4m	Day,1.8m	Night,1.4m	Night,1.8m
PSNR(db)	14.64	12.39	10.728	10.259
OCR Accuracy(%)	100	97	95	94

Table 1: Performance in Controlled Environment

Environment	Day,1.4m	Day,1.8m	Night,1.4m	Night,1.8m
PSNR(db)	13.32	11.17	9.744	9.558
OCR Accuracy(%)	100	97	92	85

Table 2: Performance in Random Environment

Table 1). Moreover, the ultimate goal of our system is the readability of the recovered images, so we also used Optical Character Recognition(OCR) services to evaluate the accuracy. On average the model achieves 96.5% OCR accuracy. Considering the complexity of Chinese characters, and the assist of context when read by a human being, we believe this result proves that this model is highly functional when ignoring the environmental parameters.

5.2 Performance In Random Environment

We train the model with data captured in different environments, as mentioned before, and test its ability in other environments(see Table 2). The model achieves 93.5% accuracy.

We compare the results to discover the influence of the surrounding environment: The model performs best in daytime and weaker illumination may decrease its ability. Also, the model performs better at closer distances, but this difference is trivial.

5.3 Adapting Ability

We train the model with fewer groups of data, exposing it to fewer variations of environment parameters, and examine the model's performance in other environments. The results are shown in Table 3.

We observed that variations in light and angle parameter in training data is crucial to a robust model. Variations in distance is not so influential to the results, given that the distances are between 1 and 2 meters. Distance changes do not have such a large impact on the size of the characters shown in the images, so that features extracted from a fixed distance might still exist when these characters vary slightly in size. However, if the network is not exposed to angled images during the training process, the rotations and deformations caused by these angles will easily disturb the feature extraction process.

5.4 Comparison with other architectures

We train and test other commonly used architectures with the same sets of data and evaluate their results. We chose SRCNN, a commonly used single image SR network, and applied it to each single image before merging the results by pixel-level averageing. We also used a multi-frame version of CNN consisting of 3D convolutional layers, designed for video super resolution(VideoSR). However, as mentioned above, it is very difficult for the single image approaches to utilize information and distinguish the noized and deformed patterns, while VideoSR approaches rely apon consistency between frames, so they fail to give satisfactory results. We used the relatively easys 'daytime 1.2m-distance direct' group of data for testing. The results are shown in Table 4.

6 SYSTEM EVALUATION

6.1 Accuracy

We build the system on smartphone and evaluate its performance in real-life environments. We experiment with a Redmi 6A smartphone (with a camera of 13 million pixels) for the attacker and a HUAWEI Mate8 smartphone for the victim. We ask 5 human participants to read the reconstructed characters to evaluate the usability of our model. No participants can read the unprocessed images, but all of them can decipher the information on the reconstructed image without much difficulty. The results are shown in Table 5.

6.2 Influence of hand tremors

We ask 5 participants to capture images with handheld smartphones, keeping their hand still to their greatest effort(Handheld camera). We process these images and let them read the results. We compare this performance to the data collected on stationary phones to evaluate the influence of hand tremors. We also ask participants to hold a smartphone in their hands and read a piece of text, without other additional instructions(Handheld target). The user may freely interact with the phone when reading. We capture images of the phone at the same time to see how our system deal with a moving target screen.The results are shown in Table 6.

We conclude from the results that hand tremors can impact the performance of our system. Although the edges of the phone are notable marks for image alignment, small shifts in the sub-pixel level will cause blurriness in the results of the networks, lowering the readability of the outputs.

6.3 Success rate in different tasks

We test the success rate of obtaining crucial information when the observed participant perform several tasks on a phone: reading text message, typing text message, entering PIN, and typing password with numbers, English and special characters. The observed participant will turn off the screen of his phone as soon as he/she finishes the task, while the observing participant will observe through our APP, constantly capturing images and processing them to display a real-time and magnified view of the observed phone. For the first two tasks,we ask the observing participant several questions to test if he/she have collected the vital information(e.g. the name or the location mentioned in the text). For the task of recognizing PIN and passwords, we use accuracy per character as a supplementary evaluation metric. The results are shown in Table 7.

We prove from this experiment that our system functions normally in everyday scenarios and poses a threat to screen privacy.

6.4 Perceived shoulder surfing susceptibility

We ask the observed participant to rate the perceived shoulder-surfing susceptibility in these scenario. The attacker will be sitting or standing behind the participant at 1.5m range, pretending to be interacting with their own phone while continuously running the shoulder-surfing APP. None of the participants were alerted by the attacker's behavior and reported suspicion of shoulder-surfing. We

Data	All Data	Night only	1.2m distance only	Direct angle only
PSNR(db)	13.32	9.17	9.94	8.84
OCR Accuracy	100	73	80	52

Table 3: Adapting Ability

Model	Ours	SRCNN	VideoSR
PSNR(db)	13.32	7.690	8.403
OCR Accuracy(%)	100	10	23

Table 4: Comparison with other architectures

believe that our system can enable a malicious attacker to gather large amounts of critical information from the victim while remaining unnoticed.

7 LIMITATIONS

7.1 Image Capturing Ability

Latest models of smartphones can capture images at 10 frames per second in burst mode easily, however, this ability is not common in models 3 4 years old. Heavily used phones also performs less than ideal when capturing images in burst mode. Also, when capturing images the user need to hold their phone as still as possible to avoid motion blur and assist image alignment. The user also has to keep a sharp focus on the target phone.

7.2 Processing Ability

To achieve best performance the user needs a phone with strong processing capabilities to run the neural network at real time. As neural networks have been common place in numerous modern APPs, most phones of the latest generation have upgraded their processing ability to run neural networks, but older versions might not possess such processing powers and cannot process images at real-time.

7.3 Motion and Tremors

In our experiment we assume the observed user will hold still his/hers phone, and not making interactions too often. However some users may tilt their phones during interactions—especially when typing with one hand. Too much tremor will cause motion blur in captured images and misalignment between frames, thus degrading the result.

8 RELATED WORK

8.1 Shoulder Surfing

With the arrival of the information era, privacy issues are becoming increasingly prominent. Smartphone screen privacy, the concern of our smartphones being observed by strangers in public areas, or shoulder surfing, have been studied heavily in recent years. Surveys of shoulder surfing provides evidence of this behavior in real world [7] [9], often more efficient than expected[14]. Various techniques and systems have been designed to mitigate this threat. Some systems sense malicious passers-by and hide information [1] or warn the user[22]; others modify the user interfaces to create honeypots (for passwords)[3], confuse unauthorized parties[27], or making the interactions invisible[13] or unreadable from a distance[4]. Most of the works designing defenses against shoulder surfing assume that the attacker is a casual passer-by, taking occasional peeks with his/hers naked eye, as is the case most of the time[7]. However, it is the malicious ones, although few, that can utilize the information (passwords, business correspondence, etc.) they obtained to do real harm. On the other hand, the threat model of tool-assisted shoulder surfing is also studied. Schaub, et, al. observed the susceptibility of shoulder surfing from the naked eye[24]. Maggi, et al. designed an automatic shoulder surfing threat model, observing the target smartphone with a camera[18], but without the help of SR techniques, this method can only function when the attacker is standing at close range, which is a barely practical scenario. With the development of smartphone camera, processing ability, and SR technology over the years, we propose a stronger shoulder surfing system in which attackers can successfully obtain information while keeping a distance to avoid suspicion, and prove that this threat is eminent in our daily life.

8.2 Super Resolution

Image super-resolution is the process of reconstructing an image with a higher spatial resolution. Single image super-resolution techniques accept a single low-res image as input, and based on its structural pattern, self-similarity [26], or previous knowledge of the genre of the image, deduces missing information and reconstructs the missing pixels to form a sharp, high-res image. On the other hand, multiple image super-resolution techniques work on a set of pictures on the same scene, e.g. multiple snapshots captured by the camera of a smartphone, successive images from a satellite, or adjacent frames on a video clip. Although they picture the same scene and are mostly identical to each other, those low-res pictures can be viewed as replicas of a certain high-res scene with random pixels blurred and removed, and multiple image super-resolution algorithms reconstruct the high-res scene by merging these incomplete information sources, often exhibiting better performance than single image super-resolution.

As mentioned above, multiple image super-resolution algorithms are based on the assumption that all input images are reflections of the same scene. It can be assumed that there exists a high-res image H, and each input low-res image Li is a version of an aliased, blurred, downsampled and noised H. By modelling the alias, blur and noise effects, and comparing these low-res inputs, we can grasp an estimation of those degrading effects and neutralize them, and the output of our super-resolution algorithm will be a high-res image H' with the most possibility to single-handedly generate all the input images Li via the aliasing, blurring, downsampling and noising processes.

Over the last two decades a variety of techniques have been proposed for the multiple image super-resolution problem. Early

Scenario	Home	Transport	Theater
Valid images	20	20	20
OCR Accuracy(%)	95	80	65
Human Recognition Accuracy(raw data)(%)	5	0	0
Human Recognition Accuracy(processed image)(%)	95	85	70

Table 5: Accuracy in different real-life scenarios

Scenario	Stationary	Handheld camera	Handheld target	Handheld camera and target
Valid images	20	19	20	18
OCR Accuracy(%)	95	85	80	80
Human Recognition Accuracy(raw data)(%)	5	0	5	5
Human Recognition Accuracy(processed image)(%)	95	85	80	85

Table 6: Influence of hand tremors

Scenario	Read text	Type text	Enter PIN	Enter complex password
Human Recognition Accuracy(raw image)	5	0	0	0
Accuracy	100	100	-	-
Accuracy per character	-	-	100	80

Table 7: Success rate in different tasks

works focus on the analysis of the images on frequency or special domains, focusing on their pixel level differences and merging their information. Spatial domain methods mostly work on the interpolation approach, trying to insert new pixels between pixels of the low-res images. One of the well-known methods in the spatial domain is the Shift-Add algorithm [8], which functions by maximizing the pixel-wise possibility of high-res image generating low-res images with gradient decent methods. On the other hand, frequency domain approaches focus on the Fourier transform of the images, and reconstruct high-res images based on patterns of the images in the frequency domain. Downsampling processes remove high-frequency components of the images, preserving low-frequency components; Noises often exists in a certain frequency band and can thus be identified and removed. By assuming that the high-res scene is band-limited, with CFT and DFT transformations, frequency domain algorithms can reconstruct high-frequency components from patterns of the low-frequency components, removing bands of noise and blur effects at the same time. A great variety of techniques have been proposed, however, due to the nature of the problem, these algorithms are all tailored for their application: natural photos, scanned text, satellite imaging, biometrics, etc. and achieve best performance only in their own field.

More recent works of Super Resolution are often based on deep learning approaches. In 2016, SRCNN [5] was developed from CNN replacing pooling layers with upsampling layers, achieving notably better performance than previous approaches. This work focuses on single image super-resolution, as the neural network accepts a single image as input. As Generative Adversarial Networks (GAN) were proposed, the network good at constructing images pleasant to the human eye was also introduced into the domain of super resolution [15]. These networks generate photo-realistic images, with less artifacts and more genuine-looking details pleasing to the human eye, though not always accurate against the real scene. Note that super resolution is an ill-posed problem, that is, we do not have a 'ground truth' to begin with, and traditional evaluation methods, e.g. mean square error (MSE) might not be suitable according to the application. Although SRGAN might score slightly lower on MSE, it can reconstruct more high-frequency and realistic details and score much higher on the mean opinion score (MOS). However, when published, the above works were limited to single input images. Because of increasing complexity and training difficulty, we cannot directly modify these super resolution neural networks to accept multiple images as input. There are also various works adapting neural networks to perform multi-image SR tasks. The most common variation is video SR[11, 25], accepting complete video clips as input, and functioning on the similarity between frames. The sequentiality and consistency between adjacent frames makes it easy for the convolution networks to comprehend, and by modifying the 2d convolution layers to 3d convolution[2], modifying the dataflow to merge neighboring frames among the network layers[10], or recurrently processing the frames under the guidance of the output of the previous frame[23], the task can be completed easily. Other works face image groups without consistency or sequential information, e.g. satellite images. To achieve appreciable results, most works choose hybrid methods, solving the multi-image SR problem with multiple single-image SR procedures. They commonly function by merging the results of single image SR algorithms to efficiently utilize input information[12], or building a multi-image network to create a course view and support it with single image SR networks[6]. These methods expect at least some degree of information to be extracted from each of the single frames with single image SR methods.

However, the methods mentioned above are not suitable for our application and we need a novel approach. The blurriness of our photos exceeds the processing ability of non-learning methods; furthermore, these photos are extremely blurred, defocused and full of noise so that few information can be extracted from them via single image SR methods; The adjacent frames display drastically different blurring patterns so that they lack consistency between adjacent frames. Only from an overall view of all the images can we distinguish noise from facts, and we need a network that can comprehend the similarities and differences between each frame to recover the buried information, and indeed, building an end-to-end multi-image SR network without assuming sequential consistency between frames is still a very much underexplored field of study, and our work aims at filling that gap.

REFERENCES

[1] F. Brudy, D. Ledo, and S. Greenberg. Is anyone looking? mediating shoulder surfing on public displays (the video). In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 159–160. 2014.

[2] J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang, and W. Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4778–4787, 2017.

[3] N. Chakraborty and S. Mondal. Tag digit based honeypot to detect shoulder surfing attack. In *International Symposium on Security in Computing and Communication*, pages 101–110. Springer, 2014.

[4] C.-Y. D. Chen, B.-Y. Lin, J. Wang, and K. G. Shin. Keep others from peeking at your mobile device screen! In *The 25th Annual International Conference*, 2019.

[5] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[6] Z. Dong, S. Zhang, B. Ma, D. Qi, L. Luo, and M. Zhou. A hybrid multi-frame super-resolution algorithm using multi-channel memristive pulse coupled neural network and sparse coding. In *2019 7th International Conference on Information, Communication and Networks (ICICN)*, pages 185–190, 2019.

[7] M. Eiband, M. Khamis, E. Von Zezschwitz, H. Hussmann, and F. Alt. Understanding shoulder surfing in the wild: Stories from users and observers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4254–4265, 2017.

[8] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Robust shift and add approach to superresolution. In *Applications of Digital Image Processing XXVI*, volume 5203, pages 121–130. International Society for Optics and Photonics, 2003.

[9] W. Goucher. Look behind you: the dangers of shoulder surfing. *Computer Fraud & Security*, 2011(11):17–20, 2011.

[10] Y. Huang, W. Wang, and L. Wang. Video super-resolution via bidirectional recurrent convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):1015–1028, 2017.

[11] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos. Video super-resolution with convolutional neural networks. *IEEE Transactions on Computational Imaging*, 2(2):109–122, 2016.

[12] M. Kawulok, P. Benecki, S. Piechaczek, K. Hrynczenko, D. Kostrzewa, and J. Nalepa. Deep learning for multiple-image super-resolution. *IEEE Geoscience and Remote Sensing Letters*, 2019.

[13] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd. Reducing shoulder-surfing by using gaze-based password entry. In *Proceedings of the 3rd symposium on Usable privacy and security*, pages 13–19, 2007.

[14] T. Kwon, S. Shin, and S. Na. Covert attentional shoulder surfing: Human adversaries are more powerful than expected. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(6):716–727, 2013.

[15] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[16] A. Lucas, S. Lopez-Tapia, R. Molina, and A. K. Katsaggelos. Generative adversarial networks and perceptual losses for video super-resolution. *IEEE Transactions on Image Processing*, 28(7):3312–3327, 2019.

[17] J. Lyn and S. Yan. Image super-resolution reconstruction based on attention mechanism and feature fusion. *arXiv preprint arXiv:2004.03939*, 2020.

[18] F. Maggi, A. Volpatto, S. Gasparini, G. Boracchi, and S. Zanero. Poster: Fast, automatic iphone shoulder surfing. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 805–808, 2011.

[19] H. Nasrollahi, K. Farajzadeh, V. Hosseini, E. Zarezadeh, and M. Abdollahzadeh. Deep artifact-free residual network for single-image super-resolution. *Signal, Image and Video Processing*, 14(2):407–415, 2020.

[20] A. Papadopoulos, T. Nguyen, E. Durmus, and N. Memon. Illusionpin: Shoulder-surfing resistant authentication using hybrid images. *IEEE Transactions on Information Forensics and Security*, 12(12):2875–2889, 2017.

[21] Polotiko. Aguirre furious at photo leak of private text message, 2017.

[22] A. Saad, M. Chukwu, and S. Schneegass. Communicating shoulder surfing attacks to users. In *Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia*, pages 147–152, 2018.

[23] M. S. Sajjadi, R. Vemulapalli, and M. Brown. Frame-recurrent video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6626–6634, 2018.

[24] F. Schaub, R. Deyhle, and M. Weber. Password entry usability and shoulder surfing susceptibility on different smartphone platforms. In *Proceedings of the 11th international conference on mobile and ubiquitous multimedia*, pages 1–10, 2012.

[25] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[26] N. Suetake, M. Sakano, and E. Uchino. Image super-resolution based on local self-similarity. *Optical review*, 15(1):26–30, 2008.

[27] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget. Design and evaluation of a shoulder-surfing resistant graphical password scheme. In *Proceedings of the working conference on Advanced visual interfaces*, pages 177–184, 2006.

[28] B. Wronski, I. Garcia-Dorado, M. Ernst, D. Kelly, M. Krainin, C.-K. Liang, M. Levoy, and P. Milanfar. Handheld multi-frame super-resolution. *ACM Transactions on Graphics (TOG)*, 38(4):1–18, 2019.